Opponent Modeling and Spatial Similarity to Retrieve and Reuse Superior Plays

Kennard Laviers[1], Gita Sukthankar[1], Matthew Klenk[2], David W. Aha[2], Matthew Molineaux[3]

[1] School of EECS, University of Central Florida, Orlando, FL,
{klaviers, gitars}@eecs.ucf.edu
[2] NCARAI, Naval Research Laboratory, Washington, DC,
{matthew.klenk.ctr, david.aha}@nrl.navy.mil
[3] Knexus Research, Springfield, VA,
matthew.molineaux@knexusresearch.com

Abstract. Plays are sequences of actions to be undertaken by a collection of agents, or teammates. The success of a play depends on a number of factors including, perhaps most importantly, the opponent's play. In this paper, we present an approach for online opponent modeling and illustrate how it can be used to improve offensive performance in the Rush 2008 football simulator. In football, team behaviors have an observable spatio-temporal structure, defined by the relative physical positions of team members over time. We demonstrate that this structure can be exploited to recognize football plays at a very early stage. Using the recognized defensive play, knowledge about expected outcomes, and spatial similarity between offensive plays, we retrieve an offensive play from the case base. This play is then (partially) reused to improve an in-progress offensive play. We call this process a *play switch*. Empirical results indicate that spatial similarity is central to play retrieval, and that substituting only a subset of the current play yields greater improvement over a full play substitution.

1 Introduction

To succeed at American Football, a team must be able to successfully execute closely-coordinated physical behavior. Teams rely on pre-existing sets of offensive and defensive plays, or *playbooks*, to achieve this coordinated behavior. By analyzing play history, it is possible to glean critical insights about future plays. In American Football, quarterbacks frequently call *audibles*, changes of play based on an assessment of the opponent's play. This task involves identifying the opponent's play and then selecting a new play for the offensive team.

In physical domains (military or athletic), team behaviors often have an observable spatio-temporal structure, defined by the relative physical positions of team members. This structure can be exploited to perform behavior recognition on traces of agent activity over time. This paper describes a method for recognizing defensive plays from spatio-temporal traces of player movement in the Rush

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE
Opponent Modeling and Spatial Similarity to Retrieve and Reuse Superior Plays

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Knexus Research Corp,9120 Beachway Lane,Springfield,VA,22153

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
In S.J. Delany (Ed.) Case-Based Reasoning for Computer Games: Papers from the ICCBR Workshop (Technical Report 7/2009). Tacoma, WA: University of Washington Tacoma, Institute of Techology

14. ABSTRACT
Plays are sequences of actions to be undertaken by a col- lection of agents, or teammates. The success of a play depends on a number of factors including, perhaps most importantly, the opponent's play. In this paper, we present an approach for online opponent model- ing and illustrate how it can be used to improve o ensive performance in the Rush 2008 football simulator. In football, team behaviors have an observable spatio-temporal structure, de ned by the relative physical po- sitions of team members over time. We demonstrate that this structure can be exploited to recognize football plays at a very early stage. Using the recognized defensive play, knowledge about expected outcomes, and spatial similarity between o ensive plays, we retrieve an o ensive play from the case base. This play is then (partially) reused to improve an in-progress o ensive play. We call this process a play switch. Empirical results indicate that spatial similarity is central to play retrieval, and that substituting only a subset of the current play yields greater improvement over a full play substitution.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

2008 Football Simulator. Rush 2008 simulates a modified version of American Football and was developed from the open source Rush 2005 game [1].

Using knowledge of play histories, we present a method for executing a *play switch* based on the potential of other plays to improve the yardage gained and their similarity to the current play. From a case-based reasoning perspective [2], this involves retrieving a superior play and adapting it to the current situation. In retrieving a superior play, we show that considering the relative similarity of the current play compared with the candidate play improves performance. Furthermore, we show that limiting the play switch to a subgroup of players is preferable to switching them all.

We begin by describing the Rush Football simulator. Next we describe our play switching approach with a detailed discussion of opposing play recognition, play similarity, and play adaptation. We outline the system that implements these ideas and present an empirical evaluation. We close with related and future work.

2 Rush Football

Football is a contest of two teams played on a rectangular field that is bordered on lengthwise sides by an end zone. Unlike American Football, Rush teams have only 8 players on the field at a time out of a roster of 18 players. The field is 100 yards by 63 yards. The game's objective is to out-score the opponent, where the offense (i.e., the team with possession of the ball), attempts to advance the ball from the line of scrimmage (i.e., the starting position of the ball) into their opponent's end zone. Therefore, an offensive play's success can be measured by the number of yards gained. Offensive plays contain the following positions:

Quarterback (QB): is given the ball at the start of each play, and will initiate either a run or pass to a receiver.

Running back (RB): begins behind the quarterback. The running back is eligible to receive a handoff or pass from the quarterback.

Fullback (RB): serves the same purpose as the RB.

Wide receiver (WR): executes passing routes and is the primary receiver for pass plays.

Offensive lineman (OL): is responsible for preventing the defense from reaching the ball carrier.

Tight end (TE): serves either as a lineman or as a receiver.

A Rush play is composed of (1) a starting formation and (2) instructions for each player in that formation. A formation is a set of (x,y) offsets from the center of the line of scrimmage. By default, instructions for each player consist of (a) an offset/destination point on the field to run to, and (b) a behavior to execute when they get there. Play instructions are similar to a conditional plan and include choice points where the players can make individual decisions as well as pre-defined behaviors that the player executes to the best of their physical capability. Rush includes three offensive formations (power, pro, and split) and

four defensive formations (23, 31, 2222, 2231). Each formation has eight different plays (numbered 1-8) that can be executed from that formation. Offensive plays typically include a handoff to the running back/fullback or a pass executed by the quarterback to one of the receivers, along with instructions for a running pattern to be followed by all the receivers. Defensive plays direct players to certain areas or toward individual offensive players with the goal of tackling the offensive player with the ball.

3 Offensive Play Switches

In American Football, the quarterback often dynamically changes the play based on the defensive formation and their reactions to offensive actions before the beginning of the play. Although Rush does not allow for actions before the play, the Rush simulator allows us to alter the play shortly after it has begun.

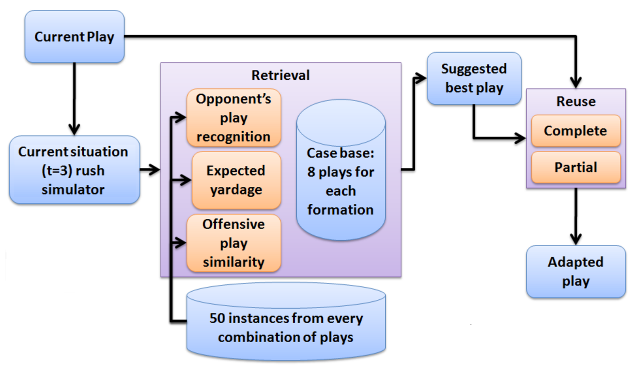


Fig. 1. Play-switching approach.

Our approach focuses on two aspects of case-based reasoning: retrieval and reuse [2]. At this early stage, we are not concerned with the revision or retention of play-switching episodes for future use. Our play switch approach is summarized in Figure 1. Our retrieval method selects an expected best offensive play by quickly recognizing the opponent's play, predicting the results of different offensive plays against it, and computing similarities between each offensive plays and the current situation. The retrieved play is reused by giving new actions to players in the current situation. Retrieval is performed using a case base of 24 plays (i.e., 8 plays for each of the three offensive formations).

The system's background knowledge includes 50 instances of every offensive and defensive play combination. These instances are used to train the recognition system, generate an expected yardage table for every combination of plays, and compute similarity between the offensive plays. The next sections describe the play recognition and similarity metric used in retrieval, followed by a discussion of how the retrieved play is adapted for the current situation.

3.1 Play Recognition using SVMs

Given a series of observations, our goal is to recognize the defensive play as quickly as possible in order to maximize our team's ability to intelligently respond with the best offense. Thus, the observation sequence grows with time unlike in standard offline activity recognition where the entire set of observations is available. We approach the problem by training a series of multi-class discriminative classifiers, each of which is designed to handle observation sequences of a particular length. In general, we expect that the early classifiers will be less accurate since they are operating with a shorter observation vector and because the positions of the players have deviated little from the initial formation.

We perform this classification using support vector machines [3]. Support vector machines (SVM) are a supervised algorithm that can be used to learn a binary classifier; they have performed well on a variety of pattern classification tasks, particularly when the dimensionality of the data is high (as in our case). Intuitively an SVM projects data points into a higher dimensional space, specified by a kernel function, and computes a maximum-margin hyperplane decision surface that separates the two classes. Support vectors are those data points that lie closest to this decision surface; if these data points were removed from the training data, the decision surface would change. More formally, given a labeled training set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in \Re^N$ is a feature vector and $y_i \in \{-1, +1\}$ is its binary class label, an SVM requires solving the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

constrained by:

$$\begin{aligned} y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + \mathbf{b}) &\geq 1 - \xi_i, \\ \xi_i &\geq 0. \end{aligned}$$

The function $\phi(.)$ that maps data points into the higher dimensional space is not explicitly represented; rather, a *kernel* function, $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)\phi(\mathbf{x}_j)$, is used to implicitly specify this mapping. In our application, we use the popular radial basis function (RBF) kernel:

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2), \gamma > 0.$$

Several extensions have been proposed to enable SVMs to operate on multi-class problems (with k rather than 2 classes), such as one-vs-all, one-vs-one, and error-correcting output codes. We employ a standard one-vs-one voting scheme where all pairwise binary classifiers, $k(k-1)/2 = 28$ for every multi-class problem in our case, are trained and the most popular class is selected. Many efficient implementations of SVMs are publicly available; we use LIBSVM [4].

We train our classifiers using a collection of simulated games in Rush collected under controlled conditions: 40 instances of every possible combination of offense (8) and defense plays (8), from each of the 12 starting formation configurations. Since the starting configuration is known, each series of SVMs is only trained with data that could be observed starting from its given configuration. For each configuration, we create a series of training sequences that accumulates spatio-temporal traces from $t = 0$ up to $t \in \{2, \ldots, 10\}$ time steps. A multiclass SVM (i.e., a collection of 28 binary SVMs) is trained for each of these training sequence lengths. Although the aggregate number of binary classifiers is large, each classifier employs only a small fraction of the dataset and is therefore efficient (and highly paralellizable). Cross-validation on a training set was used to tune the SVM parameters (C and γ) for all of the SVMs. Testing demonstrated near perfect recognition results, 96.88%, at $t = 3$, therefore this classifier was used to help select the most appropriate offensive play, as discussed below.

3.2 Play Similarity Metric

While knowledge about the opposing play is central to retrieving an effective offensive play, the similarity of the candidate plays to the current play estimates the feasibility of the play switch.

To calculate play similarities, we create a feature matrix for every formation/play combination based on background knowledge. The 13 features for each athlete A include max, min, mean, and median over x and y in addition to the following five special features:

FirstToLastAngle: Angle from starting point (x_0, y_0), to ending point (x_n, y_n), defined as $atan\left(\frac{\triangle y}{\triangle x}\right)$

StartAngle: Angle from the starting point (x_0, y_0) to (x_1, y_1), defined as $atan\left(\frac{y_1 - y_0}{x_1 - x_0}\right)$

EndAngle: Angle from (x_{n-1}, y_{n-1}) to the ending point (x_n, y_n), defined as $atan\left(\frac{\triangle y}{\triangle x}\right)$

TotalAngle: $\sum_{i=0}^{N-1} atan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)$

TotalPathDist: $\sum_{i=1}^{N} \sqrt[2]{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$

These features are similar to the ones used in [5] and more recently by [6] to match pen trajectories in sketch-based recognition tasks, another spatio-temporal task. Here, they are generalized for use with multi-player trajectories. Feature set F for a given play c ($c = 1...8$, represents possible play matches per formation) contains all features for each offensive player in the play and is described as:

$$\overrightarrow{F_c} = \{A_{c1} \cup A_{c2} \cup \ldots \cup A_{c8}\}$$

Using the 50 play instances from background knowledge, we compute a similarity vector V for every combination of offensive formation, offensive play,

defensive formation, and defensive play combination. This vector includes 8 entries (the computed similarities between the offensive play and the other plays from that formation). We define the similarity between plays as the sum of the absolute value of the differences (L_1 norm) between features F_{c_i} and F_{c_j}. In the evaluation section, we compare the performance of a similarity-based play switch mechanism vs. a play switching algorithm that focuses solely on the predicted defensive play.

3.3 Play Reuse

To reuse the new play in the current situation, we must adapt the current play. The most straightforward approach involves changing the entire play (i.e., each offensive player follows the new play from this time forward). An alternative strategy, *subgroup switching*, involves modifying the actions of only a small group of key players while leaving others alone. By segmenting the team in this fashion, we are able to combine two plays that had previously been identified as alike with regard to spatio-temporal data, but different in regards to yards gained. Based on our domain knowledge of football, we selected three subgroups as candidates to switch: {QB, RB, FB}, {OL, OL, OL}, and {WR, WR, TE}.

4 Improving the Offense with Play Switches

To improve offensive performance, our agent evaluates the competitive advantage of executing a play switch based on 1) the potential of other plays to improve the yardage gained and 2) the similarity of the candidate plays to the current play. Our algorithm for improving Rush offensive play has two main phases: a preprocess stage, which yields a play switch lookup table, and an execution stage, where the defensive play is recognized and the offense responds with an appropriate play switch for that defensive play. We train a set of SVM classifiers using 40 instances of every possible combination of offensive (8) and defensive plays (8), from each of the 12 starting formation configurations. This stage yields a set of models used for play recognition during the game. Next, we calculate and cache play switches using the following procedure:

1. Collect data by running the Rush 2008 football simulator 50 times for every play combination.
2. Create yardage lookup tables for each play combination. This information alone is insufficient to determine how good a potential play is for a play switch. The transition play must resemble our current offensive play or the offensive team will spend too much time retracing steps and perform very poorly.
3. Compute the similarity matrix between offensive plays for all formation/play combinations.
4. Create the final play switch lookup table based on both the yardage information and the play similarity.

To create the play switch lookup table, the agent first extracts a list of offensive plays L given the requirement $yards\,(L_i) > \epsilon$ where ϵ is the least amount of yardage gained before the agent changes the current offensive play to another. We used $\epsilon = 1.95$ based on a quadratic polynomial fit of total yardage gained in 6 tests with $\epsilon = \{MIN, 1.1, 1.6, 2.1, 2.6, MAX\}$ where MIN is small enough so that no plays are selected to change and MAX is set so that all plays are selected for change to the highest yardage play with no similarity comparison. Second, from the list L find the play most similar to our current play, and add it to the lookup table.

During execution, the offense uses the following procedure:

1. At each observation less than 4, collect movement traces for each player.
2. At observation 3, use LIBSVM with the collected movement traces and previously trained SVM models to identify the defensive play, j.
3. Access the lookup table to find $best(i, j)$ for our current play i.
4. If $best(i, j) \neq i$, Send a change order command to the offensive team to change to play $best(i, j)$.

As described in Section 3.3, our system allows for different methods of using the retrieved play. The agent can switch the play for either every offensive player or a subset.

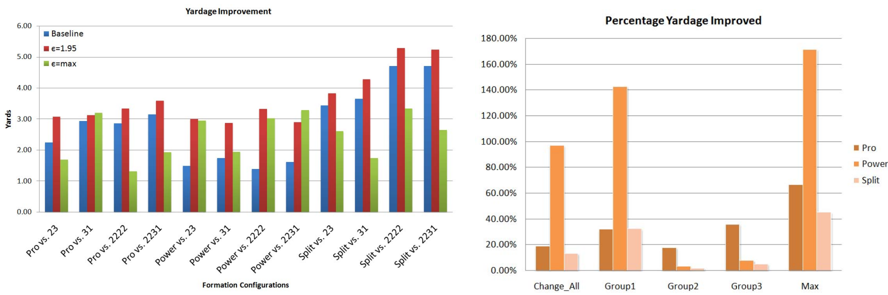
5 Empirical Evaluation

Our goal is to the answer the following questions:

1. Does our play switching algorithm improve yardage gained?
2. Does retrieval incorporating similarity with the current play outperform a greedy strategy that selects solely based upon expected yardage gained?
3. What are the effects of subgroup switching on play performance?

To answer the first two questions, we ran the RUSH 2008 simulator for ten plays on each possible play configuration under three conditions: a baseline without any play switching, our play switch model (using the yardage threshold $\epsilon = 1.95$ as determined by the quadratic fit), and a greedy play switch strategy based solely on the yardage table ($\epsilon = MAX$). The results are shown in Figure 2(a).

Overall, the average performance of the offense went from 2.82 yards per play (in the baseline condition) to 3.65 yards per play ($\epsilon = 1.95$) with an overall increase of 29%, $\pm 1.5\%$ based on sampling of three sets of ten trials. An analysis of each of the formation combinations (Figure 2(a)) shows the yardage gain varies from as much as 100% to as little as 0.1%. Power vs. 23 is dramatically boosted from about 1.5 yards to about 3 yards per play, doubling yards gained. Other combinations, such as Split vs. 23 and Pro vs. 32 already gained high yardage and improved less dramatically (i.e., about .2 to .4 yards more than the gains in the baseline sample). Overall, our model's performance is consistently better for every configuration tested.



(a) Results by play similarity (b) Results by subgroup swtiching

Fig. 2. Similarity-based switching (shown in red) outperforms both the baseline Rush offense (blue) and a greedy play switch metric (green). Changing the play for just Group 1 improves performance over changing the entire play.

Results with $\epsilon = MAX$ clearly shows simply changing to the play with greatest expected yardage generally results in poor performance. When the similarity metric is not used, the results are drastically reduced. The reason appears to be mis-coordinations between teammates accidentally introduced by the play switch; by maximizing the play similarity simultaneously, the possibility of mis-coordinations is reduced.

To evaluate the subgroup switching, we ran the simulation in three additional trails. In each trial, our play switching method was allowed to switch only one of the offensive player subgroups. Using the improvement in yardage, we compared these trials to the full offense switch and the best offensive play against the defense.

The results (shown in Figure 2(b)) clearly indicated the best subgroup switch (consistently Group 1) produced greater gains than the total team switch, which still performed better than the baseline. The Max category presents the results of an agent given the opposing play at $t = 0$, providing a ceiling. Early play recognition combined with subgroup switching yields the best results.

6 Related Work

Previous work on team behavior recognition has been primarily evaluated within athletic domains, including American Football [7], basketball [8], and Robocup soccer simulations [9–12]. In Robocup, most of the research on team intent recognition focused on coaching. Techniques have been developed to extract specific information, such as home areas [13], opponent positions during set-plays [10], and adversarial models [9], from logs of Robocup simulation league games. However, the coaching agents use offline processing to improve their team's performance in future games. In contrast, our agent immediately takes action on the recognized play to evaluate possible play switches. Ros et al. present a similar approach involving similarity between offensive and defensive alignments for

selecting plays in robocup soccer [12]. Our retrieval approach differs by using traces of player movement and a prediction concerning the opposing play. Furthermore, we demonstrate the utility of switching the play for only a subset of the offensive players. On the other hand, their representations include aspects of the overall strategy, including the score and the amount of time remaining in the game. Adding knowledge of this type is necessary for our agent to effectively play an entire football game.

Comparatively few case-based reasoning researchers have investigated spatial reasoning. Most focus on retrieving precedents based on quantitative and qualitative features [14] without any adaptation. Using insights from research on pen stroke recognition [6], our spatial similarity metric incorporates spatio-temporal knowledge into retrieval, which is then used to adapt the current situation. Galatea [15] uses stored visual problem-solving episodes consisting of visual transformations, which are employed analogically to arrive at a solution for new problems. While transfer in Galatea is iterative, our play switch is a one-shot process. Furthermore, Galatea places little emphasis on retrieval. Our model uses spatial knowledge throughout retrieval, first in categorizing the opposing team's play, then in determining the most similar play from the case base.

Rush 2008 was developed as a platform for evaluating game-playing agents and has been used to study the problem of learning strategies by observation [16]. Intention recognition has been used within Rush 2008 as part of a reinforcement learning method for controlling a single quarterback agent [17]. In this paper, our approach addresses policies across *multiple* agents.

7 Conclusion

Accurate opponent modeling is an important stepping-stone toward the creation of interesting autonomous adversaries. In this paper, we present an approach for online strategy recognition in the Rush 2008 football simulator. After identifying the defense's play, our agent evaluates the advantage of executing a play switch based on the potential of other plays to improve the yardage gained and their similarity to the current play.

We have shown that spatio-temporal features enable online strategy recognition in the early stages of a play. Furthermore, by incorporating spatial similarity into the selection of the appropriate play switch, our method avoids miscoordinations between offensive players, increasing the yardage gained. Additionally, we demonstrate that limiting the play switch to a subgroup of key players further improves performance.

In future work, we plan on extending our game playing agent to play the entire game. While our focus on gaining more yards is central to successful offense, in the complete game, offensive strategy becomes more complex, including scoring and clock management. As discussed previously, we plan to explore methods for automatically identifying key player subgroups for adapting the play by examining motion correlations between players. Finally, we plan to explore these ideas of online strategy recognition in other domains.

References

1. : Rush (2005) http://sourceforge.net/projects/rush2005/.
2. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications
3. Vapnik, V.: Statistical Learning Theory. Wiley & Sons, Inc (1998)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
5. Rubine, D.: Specifying gestures by example. Computer Graphics, Volume 25, Number 4 (1991) 329–337
6. Wobbrock, J., Wilson, D., Yang, L.: Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In: Symposium on User Interface Software and , Proceedings of the 20th annual ACM symposium on User interface software and technology. (2007)
7. Intille, S., Bobick, A.: A framework for recognizing multi-agent action from visual evidence. In: Proceedings of National Conference on Artificial Intelligence. (1999)
8. Jug, M., Pers, J., Dezman, B., Kovacic, S.: Trajectory based assessment of coordinated human activity. In: Proceedings of the International Conference on Computer Vision Systems (ICVS). (2003)
9. Riley, P., Veloso, M.: On behavior classification in adversarial environments. In Parker, L., Bekey, G., Barhen, J., eds.: Distributed Autonomous Robotic Systems 4. Springer-Verlag (2000)
10. Riley, P., Veloso, M.: Recognizing probabilistic opponent movement models. In Birk, A., Coradeschi, S., Tadorokoro, S., eds.: RoboCup-2001: Robot Soccer World Cup V. Springer Verlag (2002)
11. Kuhlmann, G., Knox, W., Stone, P.: Know thine enemy: A champion RoboCup coach agent. In: Proceedings of National Conference on Artificial Intelligence. (2006)
12. Ros, R., Veloso, M., de Mantaras, R.L., Sierra, C., Arcos, J.: Rertrieving and reusing game plays for robot soccer. In: 8th European Conference on Case-Based Reasoning (ECCBR-06). (2006)
13. Riley, P., Veloso, M., Kaminka, G.: An empirical study of coaching. In Asama, H., Arai, T., Fukuda, T., Hasegawa, T., eds.: Distributed Autonomous Robotic Systems 5. Springer-Verlag (2002)
14. Holt, A., Benwell, G.: Case-based reasoning with spatial data. In: Proceedings of the 2nd New Zealand Two-Stream International Conference on Artificial Neural Networks and Expert Systems (ANNES '95). (1995)
15. Davies, J., Goel, A., Nersessian, N.: Transfer in visual case-based problem-solving. In: Proceedings of the 6th International Conference on Case-Based Reasoning. (2005)
16. Li, N., Stracuzzi, D., Cleveland, G., Langley, P., Konik, T., Shapiro, D., Ali, K., Molineaux, M., Aha, D.: Constructing game agents from video of human behavior. In: Proceedings of AIIDE-09. (2009)
17. Molineaux, M., Aha, D., Sukthankar, G.: Beating the defense: Using plan recognition to inform learning agents. In: Proceedings of Florida Artifical Intelligence Research Society, AAAI Press (2009) 337–343